Voice Domain Name System for the Conversational Commerce Standard

Author
Esteve Tarragó Sanchis

Supervisor
Brian Subirana

AutoId Lab
Department of Mechanical Engineering
Massachusetts Institute of Technology

August 2018

Acknowledgments

I would first like to thank my thesis advisor Prof. Brian Subirana. Brian was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this paper to be my own work, but steered me in the right the direction whenever he thought I needed it.

I would also like to thank the co-workers in the Conversational Commerce Standard: Filip Cano, Alexander Jodoin and Peter Oliveira. They have been a great aid when problems arrived and they have always worked hard towards the project fulfillment. I also won't forget all the AutoId Lab members that always gave me a hand when needed, specially Yong Bin and Shane.

This extraordinary opportunity, a research stay at MIT, would not have been possible without the intervention and economic help from the Centre de Formació Interdisciplinària Superior (CFIS) and to the private foundation Cellex respectively.

My stay at Boston has been as best as possible thanks to Aleix, Maria, Martí, Laura, Miquel, Yuying and Oleguer to name a few. We shared so many grateful experiences that helped me disconnect and made me feel like at home.

Finally, I must express my very profound gratitude to my family and to my girlfriend for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Abstract

Conversational Commerce is a raising trend that will soon get a high share of the market. From the AutoID Lab we understand that there is a need for an standard in this sector.

This thesis focus on the wake-up-sentences and propose a system to regularize them by avoiding misguided sentences and creating a data and computational efficient algorithm to detect them.

Resum

El *Conversational Commerce* és un àmbit creixent que aviat obtindrà una gran part del mercat. Des del *AutoID Lab*, entenem que existeix la necessitat d'un estàndard en aquest sector.

Aquesta tesi es centra en els *wake-up-sentences* i proposa un sistema per regularitzar-les evitant frases similars i creant un algoritme eficient tant en dades com en recursos computacional per detectar-les.

Resumen

El *Conversational Commerce* es una ámbito creciente que pronto obtendrá una alta quota en el mercado. Desde el *AutoID Lab* entendemos que hay una necesidad de un estándar en este sector.

Esta tesis se centra en las *wake-up-sentences* y propone un sistema para regularizarlas, evitando oraciones similares y creando un algoritmo eficiente en datos y recursos computacionales para detectarlas.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Conversational Commerce Standard

Conversational commerce is a term which reefers to the interaction between customers and business with the aid of technology in a natural manner (i.e. in a way similar to a human conversation).

The main interfaces used nowadays to establish this interaction are voice and text. Other emerging ones are neural or glance interfaces. In this thesis we will focus on the voice interface.

One of the greatest examples of conversational commerce through voice nowadays are personal assistants as Google Now, Apple's Siri or Amazon's Alexa. All of them use a wake-up-sentence before establishing any conversation. Google and Amazon have created specific devices just to interact with their personal assistant.

Conversational commerce is a raising trend thanks to the technological development in artificial intelligence and, more particularly, deep machine learning. New hardware and software are enhancing business-costumer interactions in ways hard to imagine just a few years ago.

For the before mentioned reasons the AutoID Lab has decided to implement a standard in order to facilitate the interaction between costumers and business taking in to account privacy and transferability concerns among others.

You will be able to find more information on the topic in the Conversational

Commerce Standard white paper AutoID Lab will soon publish.

1.2 Wake Up Sentence

A wake up sentence is a set of words pronounced before establishing conversation with a machine. The computer is always listening for this sentence and once it is detected, it knows that the conversation has began. Some examples of Wake Up Sentences are:

- Google Now: *Ok, Google*
- Apple's Siri: *Siri*
- Amazon's Alexa: *Hello Alexa*

One same device can have multiple wake-up-sentences. A great example of it is Alexa that can be activated with up to 4 different wake-up-sentences.

The election of a wake-up-sentence is not easy and it's influenced by marketing and technological decisions.

1.3 Machine Learning

Machine learning is a subset of artificial intelligence in the field of computer science that uses statistical techniques to give computers the ability to "learn" with data, without being explicitly programmed.

In most of the cases you train [1] a model with existing data to make predictions on new one. The amount of data needed depends on the model complexity. As the model gets more complex, the more data is needed.

[1] Most of training algorithms use a variant of gradient decent method were the the output of the model is compared with the ideal one and the algorithm adapt the model parameters in order to increase the performance.

1.4 Deep Learning

Deep learning is a subset of *machine learning* were the models are very complex due to stacking lots of operations. This is the reason why they are called *deep* and it is normally associated with *big data* for the large amount of data needed to train the models.

Chapter 2

Voice Domain Name System

2.1 Definition

The Voice Domain Name System (VDNS) is a system that allows particulars and business to register wake-up-sentences and assign them an IP address in a similar way the DNS works. This system has to ensure that there are not names that collide nor misguided.

Sub-domains are a set of wake-up-sentences that a user could configure on it's device in order to increase accuracy and restrict the possible outpost of the conversations. Also there may exist some sponsored sub-domains which could include the most contacted ones or the ones that sponsor the project.

2.2 Use case

2.2.1 Registering a domain

Anyone may register a wake-up-sentence in the cloud. The only requirement is to have multiple recordings[1] of the activation sentence you want to register. Once you have them, you upload them to the cloud and then they are compared using the model[2] in order to ensure that they are not too similar to any other one already registered.

[1] At the AutoId Lab we are developing a tool to make it easy to anyone.
[2] Described on chapter 4.

2.2.2 Detecting a wake-up-sentence

Anyone could develop a device which is able to connect with multiple personal assistances through the Internet. All our code is open source and with low hardware requirements. It's easy to imagine a Raspberry Pi connected to your favorite brands without any intermediate in between.

Also these feature could be integrated in existing devices as Alexa or Google Home, be used from a mobile application in most of nowadays phone or from a web-browser.

Chapter 3

Previous research

Since 1950's the technology to recognize speech has been improving decade after decade. In the early 2000s, speech recognition was still dominated by traditional approaches such as Hidden Markov Models [7]. Nowadays many aspects of speech recognition have been taken over by a deep learning.

3.1 Classification

The easiest approach to wake-up-sentence detection is to use a classifier with one output node and train the neural network with audio samples labeled with *Yes* or *No*. This model requires audio of different people saying the wake-up-sentence[1] tagged as *Yes* and audio of other words, sounds and background noise tagged as *No*.

A different approach must be taken if we wish to known which word has been said among a set. What is usually done is having different nodes at the end of the neural network each representing one wake-up-sentence and an extra one representing everything else. We use the softmax [2] function to get the probability and we compare it with the one hot encoding of the class of the audio. It's important to train the network without bias to any class.

[1] It may be trained to detect more than one wake-up-sentence but you won't be able to differentiate them.

[2] Explained on chapter 5.

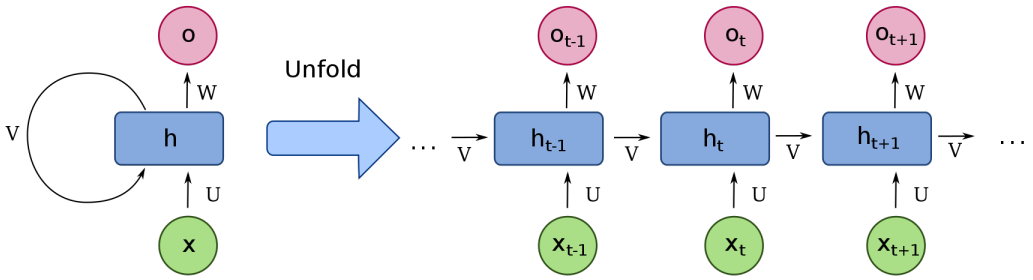


Figure 3-1: Recurrent neural network basic diagram.

3.2 Sequence to sequence

A very popular technique in speech recognition used in deep learning is sequence-to-sequence translation. The meaning of sequence-to-sequence is that the network is given a sequence as an input (a written sentence in Catalan for example) and the output is another sequence (the previous sentence translated in to English).

In the case of speech it's mostly used to translate an audio in to the corresponding transcription sentence. In order to understand how sequence-to-sequence in speech works we must first understand what recurrent neural networks and CTC are. Baidu's Deep Speech research paper[4] is a great example of it.

3.2.1 Recurrent Neural Networks

Inside most sequence-to-sequence model we can find one or multiple Recurrent Neural Networks (RNNs). RNNs are characterized by having at least one layer that transmits information to the itself in the next iteration. See figure 3-1 for a graphical explanation.

3.2.2 Connectionist Temporal Classification

Connectionist Temporal Classification (CTC) is the loss function used in speech transcription. Models trained with CTC typically use a recurrent neural network (RNN) to estimate the per time-step probabilities. An RNN usually works well since it ac-

counts for context in the input, but we're free to use any learning algorithm which produces a distribution over output classes given a fixed-size slice of the input. More details in Graves'es paper[3].

Chapter 4

The model

4.1 Objective

We designed this model having in mind to main objective. The fist one was to be used prevent collision between the registered wake-up-sentences. In order to achieve it, given two audios, the model has to classify if two words are the same (outputting 1) or they are different (outputting a 0). This will allow to register wake-up-sentences which are different to all the previous ones registered and also allow wake-up-sentence detection using the same model.

The second objective is using the same model to detect a huge amount of wake-up-sentences in real time applications on embedded devices.

4.2 Audio properties

Computers are available to store and play digital audio signals in different formats. Most formats are based on saving the intensity of sound each period. Frequency ranges are from 5000 Hz to 50000 Hz depending on the use of given to the audio.

The samplerate we have selected for our model is 16000 Hz because it is the one that's most used in deep learning and consequently the samplerate were we found more data.

4.2.1 Voice properties

The human voice speech of a typical adult have a fundamental frequency from 85 to 255 Hz. However the harmonic series, as seen in fig. 4-1, will be present for the missing fundamental to create the impression of hearing the fundamental tone.

It's important to note that when we hear a constant sound, for example the letter E repeated over time the recorded audio signal is not a constant rather a sinusoidal signal of frequency plus all it's harmonics.

A very useful tool to analyze speech recognition and that is used a by most of deep learning voice recognition tools is the spectrogram which represents the intensity of a the different sinusoidal frequencies over time. The common representation is with the x-axis being time, the y-axis being frequency and the z-axis or color being intensity.

Even further, some research[6] has shown that the logarithm of the frequency is even more useful for some representations cause humans loss information in higher frequencies. Thats the reason why our model will be using Mel-frequency cepstrums (MFCs) of the audios as input.

4.2.2 Mel-frequency cepstrum

The mel-frequency cepstrum (MFC) [1] is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency.

The difference between the cepstrum and the mel-frequency cepstrum is that in the MFC, the frequency bands are equally spaced on the mel scale, which approximates the human auditory system's response more closely than the linearly-spaced frequency bands used in the normal cepstrum.

MFCCs are commonly derived as follows:

1. Take the Fourier transform of (a windowed excerpt of) a signal.

2. Map the powers of the spectrum obtained above onto the mel scale.

3. Take the logs of the powers at each of the mel frequencies.

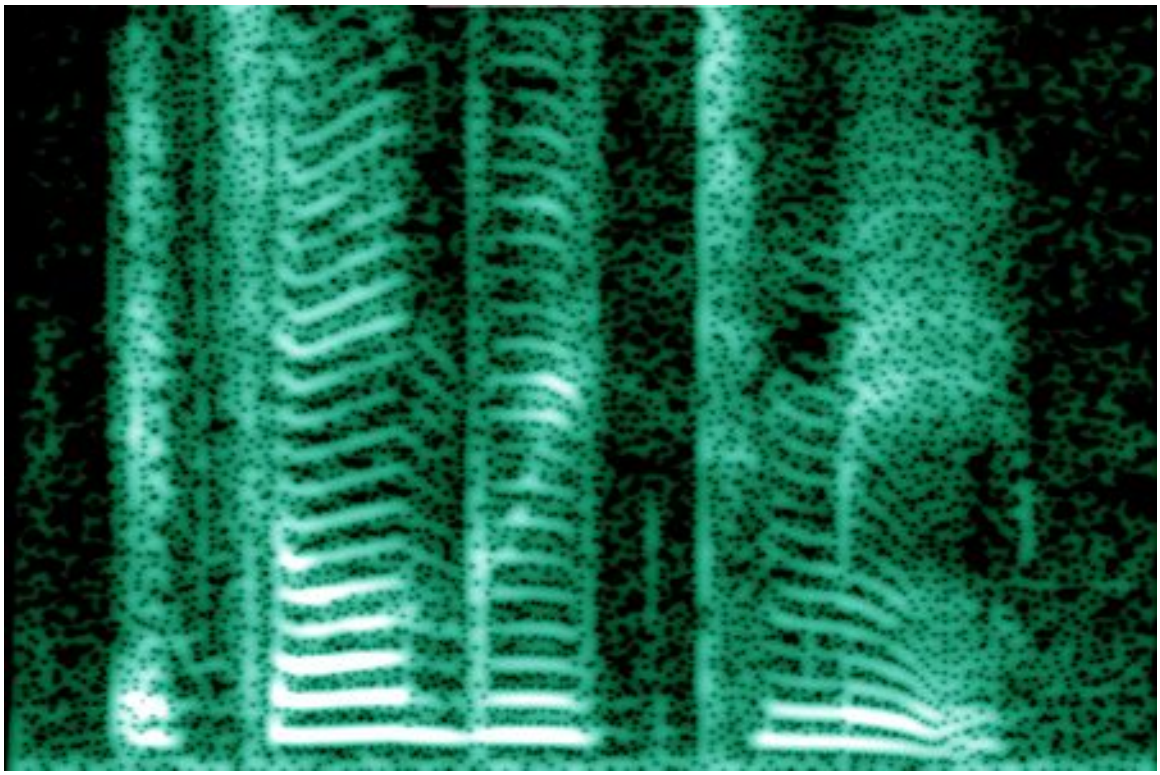


Figure 4-1: A voice spectrogram.

4. Take the discrete cosine transform of the list of mel log powers, as if it were a signal.

5. The MFCCs are the amplitudes of the resulting spectrum.

You can see and example of a MFCC in fig. 4-2.

4.3 Model properties

4.3.1 Commutativity

The first property that we can take advantage of in our problem is commutativity. Given two audios x_1 and x_2 the function should has the same output regardless to the order of the parameters.

$$D(x_1, x_2) = D(x_2, x_1)$$

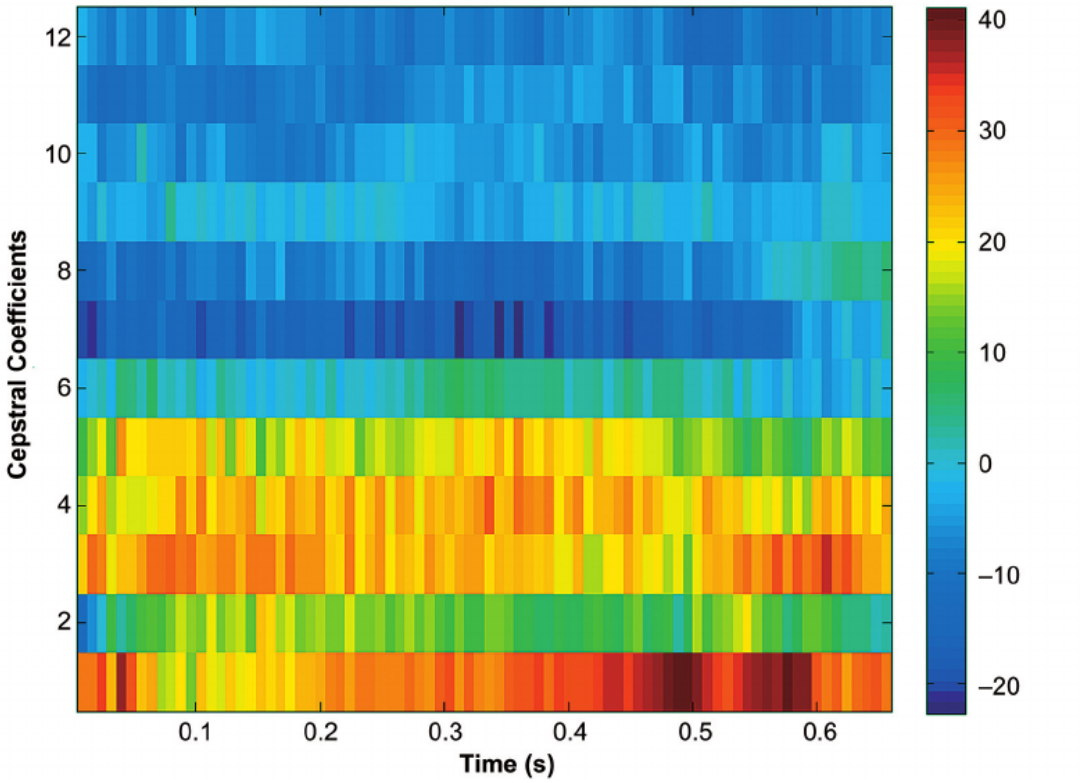


Figure 4-2: An example of a mel-frequency cepstrum.

4.3.2 Data augmentation

One property we can use to augment our labeled data is that a sound compared to itself should always has *1* as an output.

$$D(x_1, x_1) = 1$$

4.3.3 Difference

There are a lot of operations that respect commutativity. For example addition or dot products are commutative operations. Intuitively the operation we would like to perform is the distance between x_1 and x_2. The chosen operation is distance in L^1.

$$L^1 = |x_1 - x_2|$$

We have chosen L^1 in front of higher degree norm cause the gradient does not vanish when the distance is almost 0.

4.3.4 Time series domain

As the reader may have notice that $Dimemsion(x_1)$ may bot be equal to $Dimension(x_2)$. This is the reason why we will use RNN (recurrent neural networks) to create an embedding of each audio. We will use long short-term memory (LSTM) units to build the RNN.

In other words, we must be able to accept different lengths audio signals and the most efficient way to do it is iterating over the signal instead of padding it with zeros for example

4.3.5 Embedding

We will use the RNN two generate the embeddings of both audio inputs, compute the L^1 distance and then try to classify whether they are same word or not. We define embeddings as fixed lengths representations of the input audios that may allow as

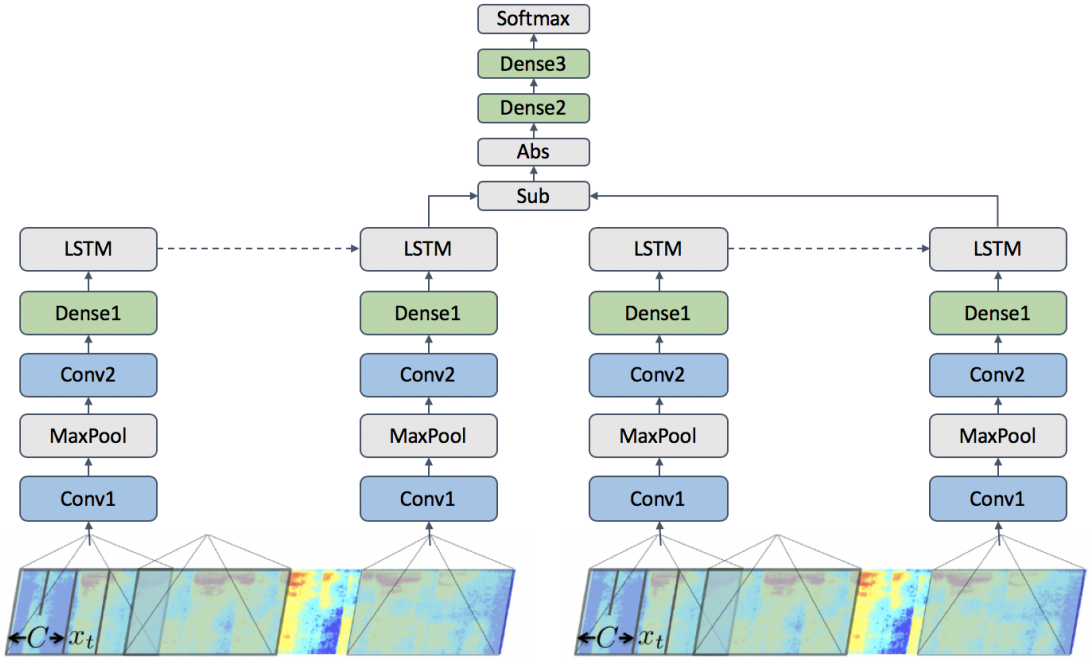


Figure 4-3: Box diagram of the model.

further optimization.

If willing to compute whether an audio x_1 is one of N words. You may have precomputed the embeddings of those N words and then just perform the L^1 distance and classification.

4.4 Model Layers

In this section we will describe the layers that compose the model. You can see a diagram of the full model in figure 4-3. Colored layers has ReLu as activation function. In table 4.1 you can see the values of each parameter of each layer used in our results.

4.4.1 Convolutional layer

Convolutional layers (figure 4-4) are very common nowadays in image processing cause they allow a more efficient detection of features all over the image. They were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex.

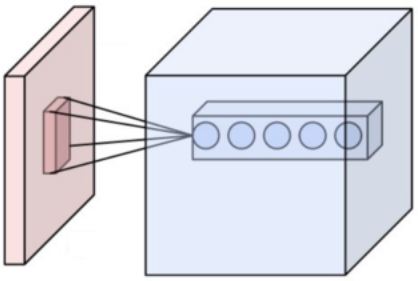


Figure 4-4: Convolutional Layer diagram.

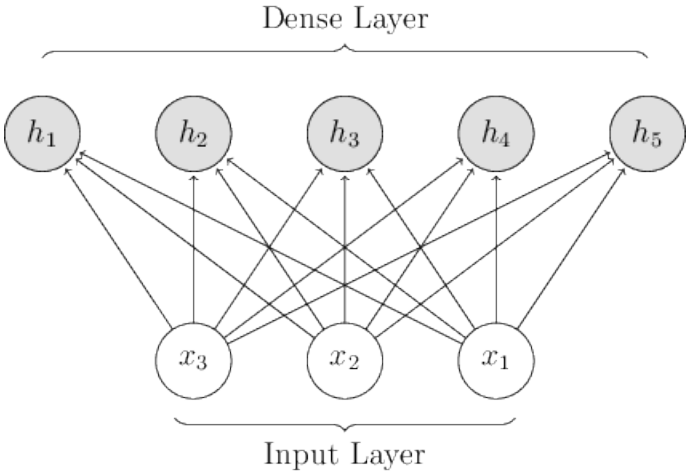


Figure 4-5: Dense Layer diagram.

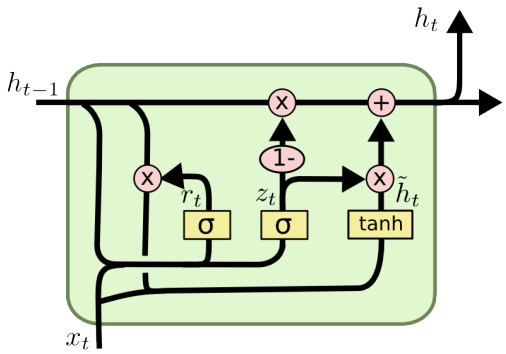
The convolution ops sweep a 2-D filter over a batch of images, applying the filter to each window of each image of the appropriate size. The filter is applied to image patches of the same size as the filter and strided in one pixel.

4.4.2 Max Pool

The pooling ops sweep a rectangular window over the input tensor, computing a max reduction operation for each window. Each pooling op uses rectangular windows of size ksize separated by offset strides.

4.4.3 Dense

A dense layer performs a matrix multiplication of the input size resizing it to the output size or number of neurons. After adding the bias term on each neuron the



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Figure 4-6: LSTM cell diagram and equations.

activation function is applied in order to achieve non-linearity. See figure 4-5 for a graphic explanation.

4.4.4 LSTM

Long short-term memory (LSTM) is a layer which is connected to itself in the next iteration. The RNN dynamics can be described using deterministic transitions from previous to current hidden states. The figure 4-6 shows a diagram and the equations of this layer.

One of the LSTM layer features is that the propagation of errors during training step are better back-propagated. This allows the model to take longer sequences avoiding the vanishing gradient[1] problem.

4.4.5 Softmax

Softmax function assigns decimal probabilities to each class in a multi-class problem. Those decimal probabilities must add up to 1.0. This additional constraint helps training converge more quickly than it otherwise would.

This equation describe how to compute the softmax probabilities:

$P(y = j \mid \mathbf{x}) = \frac{e^{\mathbf{x}^\mathsf{T}\mathbf{w}_j}}{\sum_{k=1}^{K} e^{\mathbf{x}^\mathsf{T}\mathbf{w}_k}}$

[1]The effect of multiplying n of these small numbers to compute gradients of the "first" layers in an n-layer network, meaning that the gradient (error signal) decreases exponentially with n while the first layers train very slowly.

Layer	Kernel Size	Number of kernels	Activation function	Bias
Convolutional filter #1	5 x 5	50	REctified Linear Unit	Yes
Max Pool	2 x 2	-	Maximum	No
Convolutional filter #2	5 x 5	50	REctified Linear Unit	Yes
Dense layer #1	2800 x 2048	1	REctified Linear Unit	Yes
LSTM	2048 x 2048	4	Sigmoid and Hyperbolic	Yes
Dense layer #2	2048 x 2048	1	REctified Linear Unit	Yes
Dense layer #3	2048 x 2	1	REctified Linear Unit	Yes

Table 4.1: Layers specifications.

Chapter 5

Use of the model

In order to demonstrate the use case of the model we have come up with two different use cases. There are other use cases that exist but they haven't been implemented due to lack of time. You can find them described on chapter 7.

The two cases compute the probability of an input audio being equal a set of audios previously defined. This fact allow a previous computation of this audios embeddings in order to increase the speed in application time.

5.1 Classification

Once we have the audio embeddings of the words, we compute the audio embedding of the input audio file. At this point we have all the embeddings computed and we just need to compute the central part of the model (see figure 4-3) to know the similarity

Figure 5-1: Demonstrative video of the two principal use cases.

between each word.

The word we will classify with is the one that has maximum similarity with the input. This can be use when some user wants to register a new word. If the new wake-up-word we want to register presents a high similarity with another registered word automatic registration should not be allowed and human intervention must decide if the word can be register or not due to collision. If accepted the neural network must be retrained to distinguish between this two words.

5.2 Real time wake-up-sentence detection

As in the classification method we start by computing the embeddings of a word set that we would like the device to be listening for. This step can be done in another device different than the one we want to wake up if we want to save time and just do this computation once for all the devices.

The device that's listening has to compute the microphone audio embedding if we aim to achieved locally. This can be done in real-time thanks to the RNN architecture. This method saves the RNN last state and feeds the model with the previous one and the new data from the microphone. This way we can compute the embedding without having to process more than once each audio sample. See figure 4-3 for a visual explanation.

At this point we only have to both embeddings with the two dense layers and we will obtain if the word has been said or no. This procedure does not have consistent results cause we are comparing an audio that is much longer than the once we served to the model in training time. In order to adjust this issue we propose to train the network with environmental noise and a resulting hidden state of zeros so each time the network listen to noise it is reset to its initial state.

Chapter 6

Results

In this chapter we will expose the result obtained with the model an we will compare them with other model results. All the data and the results presented here have been made on a digits dataset provided by TensorFlow for a Kaggle competition.

6.1 Training

Training is the process of feeding inputs to the network, computing the output of the network, then computing the loss and back-propagating it in order to decrease it.

The data used is 2000 spectrograms per word of 10 words *[zero, one, two, three, four, five, six ,seven, eight, nine]*. Each iteration we fit the model with a batch of 200 spectrograms pairs and weather they are the same or not. This pairs are selected such that there are the same number of even and odd pairs and equally distributed in classes.

The training process took 2h 13min and 3730 iterations. We use and early stop technique with a validation set to avoid over-fitting. The loss function used in this experiment is cross entropy [8] and the optimizer is Adam [5].

In figure 6-1 we can see how the training loss decrease over time. In comparison with figure 6-2, where we can see that at iteration 2000 the loss cease to decrease in contrast with 6-1. That is a clear sign of over-fitting. So we will stop the training at the minimum validation loss.

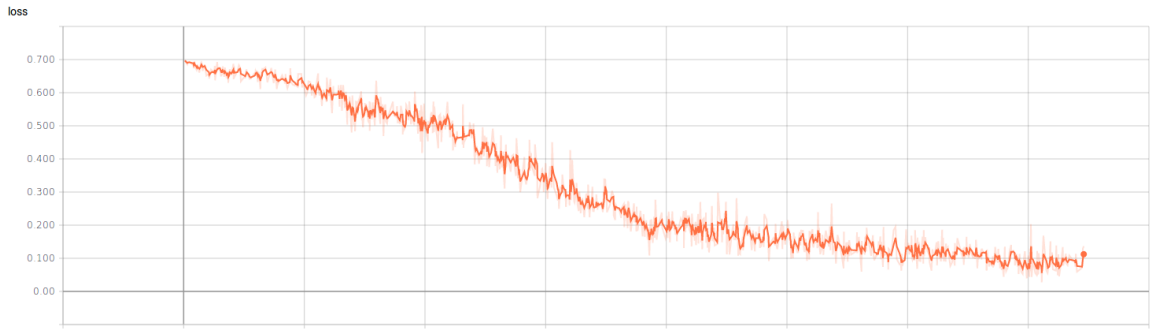


Figure 6-1: Training loss during the training.

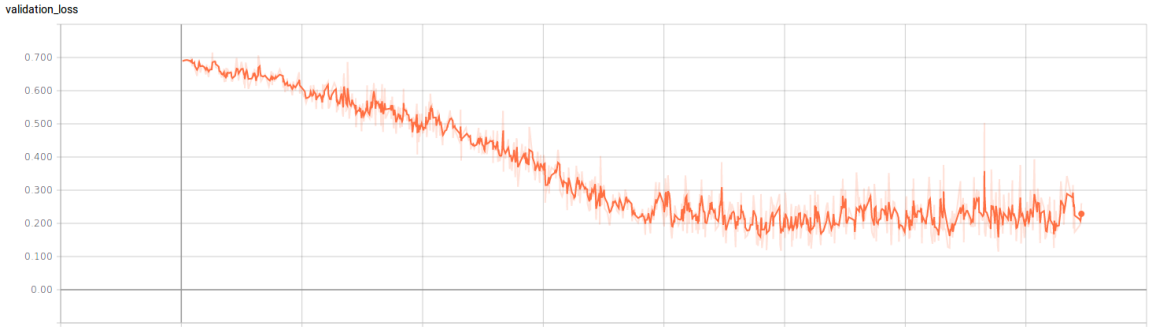


Figure 6-2: Validation loss during the training.

6.2 Results

The data was obtained from a Kaggle competition [2]. The objective of it was to classify one second words in to classes. The winner team obtained a categorization accuracy score of 91%. The vocabulary words they had to classify where *[yes, no, up, down, left, right, on, off, stop, go]*.

We obtained a score of 95% (see fig. 6-3) in the task of classifying whether te word was the same or not over digits set. We are not working under the same set or neither under the same task so we should take this comparison just as a reference. In the classification task our model achieved 85% accuracy which is not as good as other model results but ours has other advantages.

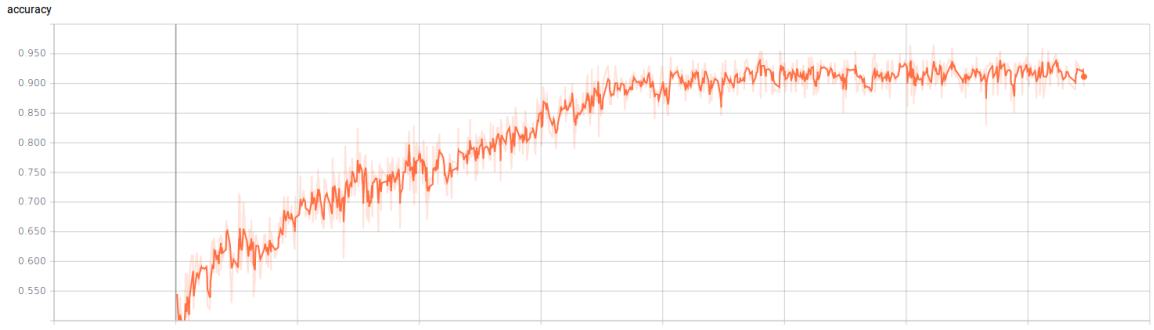


Figure 6-3: Validation accuracy during the training.

Chapter 7

Further research

In order to solve the Voice Domain Name System problem there are still some challenges missing to be faced.

7.1 Scalability

We should find a model able to scale thousands of wake-up-sentence. Currently we don't have this data so we cannot train the model. The before mentioned has a high capacity but we should make sure that it adapts to more words and some modification might be necessary.

7.2 Embedded systems

Most test have been done in different laptops. An important thing for the model is being able to run in real time on embedded systems so further testing is needed in this direction. We are developing a snap package to be deployed on IoT devices as Raspberry Pi.

7.3 Background cancellation

The model suffers to get great results when none of the wake-up-sentences is said. Thats why we are making experiments on returning to initial state when background noise or conversation is being recorded by training the embedder to return to initial position when noise is heard.

7.4 User base

One important thing for the Voice Name Domain System to work out is having a large user base. In order to achieve this we must provided lots of developer tools to make it easy to integrate the system in most of devices so VDNS could be used by the users.

Chapter 8

Conclusions

This thesis analyzes the problem of wake-up-sentences and pretend to change how most devices handle skills or applications. We propose a solution on how to solve this problem by the registration of wake-up-sentence without collisions and a light model able to quickly detect the sentences on the fly.

We belief that the Conversational Commerce should be as open as the Internet and there is no more open way than creating a VDNS to redirect queries to owners of the domains so they can handle it how they wish.

Voice domain registration is not as easy as text but this thesis stands as proof that wake-up-sentence similarity can be computed and threshold can be established. Whats even more important about this comparative model it's that allow high efficiency detection reusing most of the computations already done.

We encourage individual and enterprises start registering their voice domain and start implementing voice assistants. The technology is quickly advancing and in a not so long future developing a voice assistant won't be more difficult than a text bot.

Bibliography

[1] Comparison of MFCC and cepstral coefficients as a feature set for PCG biometric systems.

[2] TensorFlow speech recognition challenge.

[3] Alex Graves, Santiago Fernandez, Faustino Gomez, and Jurgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. page 8.

[4] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng. Deep speech: Scaling up end-to-end speech recognition.

[5] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization.

[6] Dr Ir Stéphane Pigeon. The non-linearities of the human ear.

[7] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. 77(2):257–286.

[8] J. Shore and R. Johnson. Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy. 26(1):26–37.